



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1460  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/982,243	10/16/2001	Ashish Prakash	005693.P006X	8578

48102 7590 04/19/2006

NETWORK APPLIANCE/BLAKELY  
12400 WILSHIRE BLVD  
SEVENTH FLOOR  
LOS ANGELES, CA 90025-1030

EXAMINER

CHOUDHURY, AZIZUL Q

ART UNIT PAPER NUMBER

2145

DATE MAILED: 04/19/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

**Office Action Summary**

Application No.

09/982,243

Applicant(s)

ASHISH PRAKASH

Examiner

Azizul Choudhury

Art Unit

2145

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 02 February 2006.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 21-45 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 21-45 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)  | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)                                   | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)             |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____  |

***Detailed Action***

This office action is in response to the correspondence received on February 2, 2006.

***Claim Rejections - 35 USC § 112***

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 21, 27, 34, 35, 40, 44, and 45 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The claims now possess the newly amended phrase, "relative to each other." The term "relative" is considered indefinite in the art and fails to distinctly point out to one skilled in the art, how the fields are to appear. Appropriate corrections are required.

***Claim Rejections - 35 USC § 102***

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 21-45 are rejected under 35 U.S.C. 102(b) as being anticipated by  
Huckins (US Pat No: 5,612,898).

1. With regards to claim 21, Huckins teaches a device to log information in a network cache, the device comprising: a user interface to allow a user to select a protocol, to select for logging some or all of a plurality of fields of the protocol that may be present in a messages to be received, and to specify a sequence in which the selected fields are to appear relative to each other in a log file; a first data structure for storing a value indicating a position in the user specified sequence for each selected field; a protocol independent log module to receive information from an application module, to store the information in a second data structure, and to store a reference to the information for each selected field stored in the second data structure in a location of a third data structure that corresponds to the position in the user-specified sequence for the corresponding field; and a log file wherein the information for each selected field from the second data structure is sequentially written using the reference from the third data structure (Huckins teaches a design that allows for a user to log protocol specific details (column 2, lines 21-30, Huckins). The monitoring by which the logging occurs is customizable by the user (column 2, lines 31-34, Huckins). The design also allows for the logging of only select, desired features (column 7, lines 7-23, Huckins). This includes the manner in which the monitored data is displayed (column 2, lines 31-41, Huckins). In addition, it is inherent that since data is being handled and managed, that data structures are used to store, transfer and manipulate data with. One such example of a data structure is the PDU (protocol data unit) (column 5, lines 16-40, Huckins)).

2. With regards to claim 22, Huckins teaches the device of claim 21, wherein the application module is protocol specific and obtains information for each selected field associated with the message (The Huckins design is protocol specific and provides protocol specific events (column 2, lines 21-31, Huckins). The display of the information is customizable (column 2, lines 31-41, Huckins)).
3. With regards to claim 23, Huckins teaches the device of claim 21, wherein the user interface allows the user to create new fields in addition to the plurality of fields (Huckins' design allows for the select enabling and disabling of logging of desired features (column 6, line 25 – column 7, line 23, Huckins). The data to be displayed is customizable (column 2, lines 31-41, Huckins)).
4. With regards to claim 24, Huckins teaches the device of claim 21, wherein each location in the first data structure is pre-initialized to contain a flag before the user-specified sequence is stored, the flag being utilized as an indicator that the user did not select that field for logging (The initialization of data structures is inherent. Without doing so, programs run the risk of contaminating the data stored within the data structure, during execution of the program. In addition, the Huckins' design allows for an initialization process to setup the design to monitor the designated traits (column 6, lines 12-24, Huckins)).

5. With regards to claim 25, Huckins teaches the device of claim 21, wherein the user interface is a graphical user interface (Huckins' design can be implemented in Windows™ (column 4, lines 49-61, Huckins)).
6. With regards to claim 26, Huckins teaches the device of claim 21, wherein the user interface is a command line interface (Huckins' design allows for the outputting of the data as text (column 5, lines 7-22, Huckins)).
7. With regards to claim 27, Huckins teaches a method of logging information in a network cache, the method comprising: providing a user interface to allow a user to select for logging some or all of a plurality of fields that may be present in a message to be received and to specify a sequence in which the selected fields are to appear relative to each other in a log file; storing in a first data structure a value indicating the position in the user-specified sequence of each selected field; in response to a message received over a network from a remote node, obtaining information for each selected field associated with the message and storing the information in a second data structure, in a sequence independent of the user-specified sequence, storing in a third data structure, based on the first data structure, a reference to the information for each selected field stored in the second data structure, including storing each reference in a location of the third data structure that corresponds to the position in the user-specified sequence of the corresponding field; and using the third data structure to output the

information for each selected field in the second data structure to a log file, such that the information for each selected field appears in the log file according to the user-specified sequence (Huckins teaches a design that allows for a user, who is remote to the site being monitored to log protocol specific details (column 2, lines 21-30, Huckins). The monitoring by which the logging occurs is customizable by the user (column 2, lines 31-34, Huckins). The design also allows for the logging of only select, desired features (column 7, lines 7-23, Huckins). This includes the manner in which the monitored data is displayed (column 2, lines 31-41, Huckins). In addition, it is inherent that since data is being handled and managed, that data structures are used to store, transfer and manipulate data with. One such example of a data structure is the PDU (protocol data unit) (column 5, lines 16-40, Huckins)).

8. With regards to claim 28, Huckins teaches the method of claim 27, wherein the user interface allows the user to create new fields in addition to the plurality of fields (Huckins' design allows for the select enabling and disabling of logging of desired features (column 6, line 25 – column 7, line 23, Huckins). The data to be displayed is customizable (column 2, lines 31-41, Huckins)).
9. With regards to claim 29, Huckins teaches the method of claim 27, wherein the information for each field is converted to an ASCII representation and is of

variable length (Huckins allows the data to be displayed in ASCII (column 5, lines 17-22, Huckins)).

10. With regards to claim 30, Huckins teaches the method of claim 27, wherein each location in the first data structure is pre-initialized to contain a flag before the user-specified sequence is stored, the flag to be utilized as an indicator that the user did not select that field for logging (The initialization of data structures is inherent. Without doing so, programs run the risk of contaminating the data stored within the data structure, during execution of the program. In addition, the Huckins' design allows for an initialization process to setup the design to monitor the designated traits (column 6, lines 12-24, Huckins)).

11. With regards to claim 31, Huckins teaches the method of claim 27, wherein the second data structure and the third data structure are created to respond to logging for the message and destroyed once logging for the message is completed (It is inherent that since data is being handled and managed, that data structures are used to store, transfer and manipulate data with. One such example of a data structure is the PDU (protocol data unit) (column 5, lines 16-40, Huckins). Plus, it is also inherent that memory management means (creating and deleting data structures at the appropriate times) are applied to prevent crashing a computer due to using up all the available memory).



12. With regards to claim 32, Huckins teaches the method of claim 27, wherein the first data structure persists through logging for a plurality of messages received from remote nodes (It is inherent that since data is being handled and managed, that data structures are used to store, transfer and manipulate data with. One such example of a data structure is the PDU (protocol data unit) (column 5, lines 16-40, Huckins)).
13. With regards to claim 33, Huckins teaches the method of claim 27, wherein using the third data structure to output the information further comprises sequentially accessing the third data structure to read the position of the information corresponding to each selected field and accessing the second data structure to read information corresponding to each selected field at the position indicated by the reference (It is inherent that since data is being handled and managed, that data structures are used to store, transfer and manipulate data with. One such example of a data structure is the PDU (protocol data unit) (column 5, lines 16-40, Huckins)).
14. With regards to claim 34, Huckins teaches a device for logging information in a network cache, the device comprising: a user interface to allow a user to select a protocol, to select for logging some or all of a plurality of fields of a message to be received from a remote node, the fields corresponding to the selected protocol, and to specify a sequence in which the selected fields are to appear relative to each

other in a log file; a protocol specific application module to obtain information for each selected field associated with the message; a protocol independent log module to receive information for each selected field from the protocol specific application module and to store the information for each selected field in a log file in the sequence specified by the user (Huckins teaches a design that allows for a user to log protocol specific details (column 2, lines 21-30, Huckins). The monitoring by which the logging occurs is customizable by the user (column 2, lines 31-34, Huckins). The design also allows for the logging of only select, desired features (column 7, lines 7-23, Huckins). This includes the manner in which the monitored data is displayed (column 2, lines 31-41, Huckins)).

15. With regards to claim 35, Huckins teaches a device to log information in a network cache, the device comprising: a user interface to allow a user to select for logging some or all of a plurality of fields that may be present in a message from to be received, wherein the user is allowed to specify a sequence in which the selected fields are to be subsequently output in a log file relative to each other, and wherein the user-specified sequence of each selected field is stored in a first data structure; an application module to receive the message from the remote node, to access the first data structure to determine if a field is to be logged and the sequence in which the selected fields are to be stored in the log file, and to send information for each selected field associated with the message along with the sequence number for that field to a log module; a setup and

destroy module to dynamically create and destroy a second data structure and a third data structure in response to requests from the application module; the log module to receive information corresponding with each selected field and the sequence of the information for each selected field from the application module, to store information corresponding with each selected field in the second data structure, and to store in the third data structure a reference to the position of information for each field in the second data structure, each reference being stored in a location of the third data structure corresponding to the user specified sequence of each selected field; an output module to sequentially access the third data structure to read the position of the ASCII representation of each selected field and to access the second data structure to read the ASCII representation of each selected field at the position indicated by the reference; and a log file wherein the information for each selected field from the second data structure is sequentially written using the reference in the third data structure (Huckins teaches a design that allows for a user to log protocol specific details (column 2, lines 21-30, Huckins). The monitoring by which the logging occurs is customizable by the user (column 2, lines 31-34, Huckins). The design also allows for the logging of only select, desired features (column 7, lines 7-23, Huckins). This includes the manner in which the monitored data is displayed (column 2, lines 31-41, Huckins). In addition, it is inherent that since data is being handled and managed, that data structures are used to store, transfer and manipulate data with. One such example of a data structure is the PDU (protocol

data unit) (column 5, lines 16-40, Huckins). Plus, Huckins allows the data to be displayed in ASCII (column 5, lines 17-22, Huckins). The data to be displayed is customizable (column 2, lines 31-41, Huckins)).

16. With regards to claim 36, Huckins teaches the device of claim 35, wherein the user interface allows the user to create new fields in addition to the plurality of fields (Huckins' design allows for the select enabling and disabling of logging of desired features (column 6, line 25 – column 7, line 23, Huckins). The data to be displayed is customizable (column 2, lines 31-41, Huckins)).
17. With regards to claim 37, Huckins teaches the device of claim 35, wherein each location in the first data structure is pre-initialized to contain a flag before the user-specified sequence is stored, the flag being utilized as an indicator that the user did not select that field for logging (The initialization of data structures is inherent. Without doing so, programs run the risk of contaminating the data stored within the data structure, during execution of the program. In addition, the Huckins' design allows for an initialization process to setup the design to monitor the designated traits (column 6, lines 12-24, Huckins)).
18. With regards to claim 38, Huckins teaches the device of claim 35, wherein the user interface is a graphical user interface (Huckins' design can be implemented in Windows™ (column 4, lines 49-61, Huckins)).

19. With regards to claim 39, Huckins teaches the device of claim 35, wherein the user interface is a command line interface (Huckins' design allows for the outputting of the data as text (column 5, lines 7-22, Huckins)).

20. With regards to claim 40, Huckins teaches a method of logging information in a network cache, the method comprising: providing a user interface to allow a user to select for logging some or all of a plurality of fields that may be present in a message received and to specify a sequence in which the selected fields are to appear relative to each other in a log file; storing in a first data structure a value indicating a position in the user specified sequence for each selected field; receiving a message over a network from a remote node; creating a second data structure and a third data structure to correspond to the received message; examining the first data structure to determine which fields to extract; extracting information for each selected field from the message; obtaining an ASCII representation of the information for each selected field; placing the ASCII representation corresponding with each selected field in the second data structure; placing in the third data structure a reference to the position of each ASCII representation of each selected field in the second data structure, wherein each reference is stored in a location of the third data structure corresponding to the position in the user specified sequence of the corresponding field in said subset; sequentially accessing the third data structure to read the position of the

ASCII representation of each selected field and accessing the second data structure to read the ASCII representation of each selected field at the position indicated by the reference; writing to the log file the ASCII representation of each selected field from the second data structure as each ASCII representation is sequentially accessed using the reference in the third data structure; and removing the second data structure and the third data structure (Huckins teaches a design that allows for a user to log protocol specific details (column 2, lines 21-30, Huckins). The monitoring by which the logging occurs is customizable by the user (column 2, lines 31-34, Huckins). The design also allows for the logging of only select, desired features (column 7, lines 7-23, Huckins). This includes the manner in which the monitored data is displayed (column 2, lines 31-41, Huckins). In addition, it is inherent that since data is being handled and managed, that data structures are used to store, transfer and manipulate data with. One such example of a data structure is the PDU (protocol data unit) (column 5, lines 16-40, Huckins). Plus, Huckins allows the data to be displayed in ASCII (column 5, lines 17-22, Huckins). The data to be displayed is customizable (column 2, lines 31-41, Huckins)).

21. With regards to claim 41, Huckins teaches the method of claim 40, wherein the user interface allows the user to create new fields in addition to the plurality of fields (Huckins' design allows for the select enabling and disabling of logging of

desired features (column 6, line 25 – column 7, line 23, Huckins). The data to be displayed is customizable (column 2, lines 31-41, Huckins)).

22. With regards to claim 42, Huckins teaches the method of claim 40, wherein the information for each field is converted to an ASCII representation and is of variable length (Huckins allows the data to be displayed in ASCII (column 5, lines 17-22, Huckins)).

23. With regards to claim 43, Huckins teaches the method of claim 40, wherein each location in the first data structure is pre-initialized to contain a flag before the user-specified sequence is stored (The initialization of data structures is inherent. Without doing so, programs run the risk of contaminating the data stored within the data structure, during execution of the program. In addition, the Huckins' design allows for an initialization process to setup the design to monitor the designated traits (column 6, lines 12-24, Huckins)).

24. With regards to claim 44, Huckins teaches a device for logging information in a network cache, the device comprising: means for providing a user interface to allow a user to select for logging some or all of a plurality of fields that may be present in a message to be received and to specify a sequence in which the selected fields are to appear relative to each other in a log file; means for storing in a first data structure a value indicating a position in the user-specified

sequence for each selected field; means for receiving a message over a network from a remote node; means for creating a second data structure and a third data structure to correspond to the received message; means for examining the first data structure to determine which fields to extract; means for extracting information for each selected field from the message; means for obtaining an ASCII representation of the information for each selected field; means for placing the ASCII representation corresponding with each selected field in the second data structure; means for placing in the third data structure a reference to the position of each ASCII representation of each selected field in the second data structure, wherein each reference is stored in a location of the third data structure corresponding to the user specified sequence of each field in the subset; means for sequentially accessing the third data structure to read the position in the user-specified sequence of the ASCII representation of each selected field ,and for accessing the second data structure to read the ASCII representation of each selected field at the position indicated by the reference; means for writing to the log file the ASCII representation of each selected field from the second data structure as each ASCII representation is sequentially accessed using the reference in the third data structure; and means for removing the second data structure and the third data structure (Huckins teaches a design that allows for a user to log protocol specific details (column 2, lines 21-30, Huckins). The monitoring by which the logging occurs is customizable by the user (column 2, lines 31-34, Huckins). The design also allows for the logging of only select,



desired features (column 7, lines 7-23, Huckins). This includes the manner in which the monitored data is displayed (column 2, lines 31-41, Huckins). In addition, it is inherent that since data is being handled and managed, that data structures are used to store, transfer and manipulate data with. One such example of a data structure is the PDU (protocol data unit) (column 5, lines 16-40, Huckins). Plus, Huckins allows the data to be displayed in ASCII (column 5, lines 17-22, Huckins). The data to be displayed is customizable (column 2, lines 31-41, Huckins)).

25. With regards to claim 45, Huckins teaches a computer program embodied on a computer-readable medium representing sequences of instructions which, when executed by a processor cause the processor to perform a process comprising: providing a user interface to allow a user to select for logging some or all of a plurality of fields of a message to be received, the fields dependent upon a protocol used to convey the message, and to specify a sequence in which the selected fields are to appear relative to each other in a log file; storing in a first data structure a value indicating a position in the user-specified sequence for each selected field; in response to a message received over a network from a remote node, obtaining information for each selected field associated with the message and storing the information in a second data structure, in a sequence independent of the user specified sequence; storing in a third data structure, based on the first data structure, a reference to the information for each selected

field stored in the second data structure, including storing each reference in a location of the third data structure that corresponds to the position in the user-specified sequence of the corresponding field; and using the third data structure to output the information for each selected field in the second data structure to a log file, such that the information for each selected field appears in the log file according to the user-specified sequence (Huckins teaches a design that allows for a user to log protocol specific details (column 2, lines 21-30, Huckins). The monitoring by which the logging occurs is customizable by the user (column 2, lines 31-34, Huckins). The design also allows for the logging of only select, desired features (column 7, lines 7-23, Huckins). This includes the manner in which the monitored data is displayed (column 2, lines 31-41, Huckins). In addition, it is inherent that since data is being handled and managed, that data structures are used to store, transfer and manipulate data with. One such example of a data structure is the PDU (protocol data unit) (column 5, lines 16-40, Huckins). Plus, Huckins allows the data to be displayed in ASCII (column 5, lines 17-22, Huckins). The data to be displayed is customizable (column 2, lines 31-41, Huckins)).

### ***Remarks***

The amendment received on February 2, 2006 has been carefully examined but is not deemed fully persuasive. The following are the examiner's response to the concerns expressed within the remarks portion of the amendment.

The first point of contention addressed by the applicant's representative concerns the 112-type rejection. The applicant's representative insists that the term "relative" is not indefinite when read with the remainder of the claim language.

***"...to specify a sequence in which the selected fields are to appear relative to each other..."***

The applicant's representative also suggests that possibly the examiner has confused the term "relative" with a principle that relative claim terms sometimes render claims indefinite. The examiner first would like to reassure the applicant's representative that he has not confused the term "relative" with the principle cited. In addition, the examiner would like to restate that the term "relative" is considered broad and indefinite in the art. In computer networks, the phrase "...to specify a sequence in which the selected fields are to appear relative to each other ..." makes it difficult for one skilled in the art to grasp an understanding of what type of relationship the fields are capable of having. The examiner is not requiring a specific sequence of fields. The examiner however stands by the rejection because in the art, it is difficult to ascertain within a reasonable scope, what types of relationships are possible for the fields since that is how the claims currently read.

The second point of contention involves the claim traits of selecting the protocol, selecting the particular fields of a message for logging and selecting a sequence in which the fields are to appear in a log file. Huckins teaches that protocol specific details can be logged (column 2, lines 21-30, Huckins). Plus, the user is able to customize what is logged (column 2, lines 31-34, Huckins). This includes what elements should

Art Unit: 2145

not be logged (column 7, lines 7-23, Huckins). The examiner contends that these traits within Huckins' disclosure teach the claimed selection traits.

### ***Conclusion***

**THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Azizul Choudhury whose telephone number is (571) 272-3909. The examiner can normally be reached on M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jason Cardone can be reached on (571) 272-3933. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2145

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AC

  
ZARNI MAUNG  
SUPERVISORY PATENT EXAMINER